(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 089 173 A2

(12) # EUROPEAN PATENT APPLICATION

(43) Date of publication:
04.04.2001 Bulletin 2001/14

(51) Int Cl.7: G06F 9/455, G06F 9/46

(21) Application number: 00308493.6

(22) Date of filing: 27.09.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 28.09.1999 US 407594

(71) Applicant: International Business Machines Corporation
Armonk, NY 10504 (US)

(72) Inventors:
- King, Gary M., c/o IBM United Kingdom Ltd. Winchester, Hampshire SO21 2JN (GB)
- Kubala, Jeffrey P., c/o IBM United Kingdom Ltd. Winchester, Hampshire SO21 2JN (GB)
- Nick, Jeffrey M., c/o IBM United Kingdom Ltd. Winchester, Hampshire SO21 2JN (GB)
- Yocom, Peter B., c/o IBM United Kingdom Ltd. Winchester, Hampshire SO21 2JN (GB)

(74) Representative: Davies, Simon Robert
IBM,
United Kingdom Limited,
Intellectual Property Law,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) **Dynamic adjustment of logical processor configuration**

(57) The configuration of the logical processors of a logical partition is managed dynamically. A logical partition is initially configured with one or more logical processors. Thereafter, the configuration can be dynamically adjusted. This dynamic adjustment may be in response to workload of the logical partition.

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] This invention relates, in general, to managing logical processors in a computing environment, and particularly to one including one or more logical partitions.

[0002] Logical partitioning allows the establishment of a plurality of system images within a single physical machine or central processor complex (CPC). Each system image is capable of operating as if it was a separate computer system. That is, each logical partition can be independently reset, initially loaded with an operating system that may be different for each logical partition, and operated with different software programs using different input/output (I/O) devices.

[0003] Examples of logically partitioned computing systems are described in, for instance, U.S. 4,564,903, issued on January 14, 1986; U.S. 4,843,541, issued on June 27, 1989; and U.S. 5,564,040, issued on October 08, 1996.

[0004] Commercial embodiments of logically partitioned systems include, for example, IBM S/390® processors with the Processor Resource/Systems Manager™ (PR/SM™) feature, which is described, for instance, in the IBM publication Processor Resource/Systems Manager Planning Guide, GA22-7236-04, March 1999.

[0005] One important aspect of a logically partitioned system is the management of workload running within the partitions of that system. In S/390 systems, for example, workload managers are used to manage the workload within and among the partitions. The workload managers attempt to balance the workload of the partitions by moving work to the physical resources of the system. In order to move the work, however, it is important to ensure that the data needed by the relocated work is at the moved location. This need often restricts the movement of work. Thus, it is desirable to further improve workload management in computer systems.

Summary of the Invention

[0006] Accordingly, the invention provides a method of managing logical processors of a computing environment including one or more logical partitions, said method comprising the steps of:

configuring a logical partition of said computing environment with one or more logical processors; and
dynamically adjusting the configuration.

[0007] In a preferred embodiment, said dynamically adjusting is in response to the workload of said logical partition, and may increase or decrease the number of logical processors allocated to said logical partition. A determination that said configuration is to be adjusted is preferably performed at a plurality of times, typically at regularly spaced intervals. The determination is made using a predefined equation; the result of this can be compared with one or more thresholds to determine whether the adjustment is to be made.

[0008] In the preferred embodiment, said predefined equation comprises: L=floor[max(W,U) *P+1.5], wherein: L=number of logical processors configured to said logical partition; W=percentage of central processor capacity assigned to said logical partition; U=percentage of central processor capacity currently being utilized by said logical partition; and P=number of physical processors that can be allocated on the central processor associated with said logical partition. This is subject to a maximum of L=P (ie the number of logical processor cannot exceed the number of physical processors).

[0009] The invention further provides a computer program for implementing a method such as described above. Such a program will typically be encoded on a computer usable media, which may be included as a part of a computer system or sold separately. In other words, such a computer usable media or program storage device contains program instructions executable by the machine in order to perform the methods described above.

[0010] The invention further provides a system for managing logical processors in a computing environment including one or more logical partitions, said system comprising:

means for configuring a logical partition of said computing environment with one or more logical processors; and
means for dynamically adjusting the configuration.

[0011] The dynamic adjustment of the configuration of logical processors of logical partitions in a computing environment allows the number of logical processors configured to a logical partition to remain close to the number of physical CPUs desired to provide the CPC capacity assigned to (or used by) a logical partition. Thus, the number of logical processors to manage is minimized.

[0012] Various preferred embodiments of the invention will now be described in detail by way of example only with reference to the following drawings:

FIG. 1a depicts one example of a computing environment;
FIG. 1b depicts a further embodiment of a computing environment;
FIG. 2 depicts additional components of a computing environment;
FIG. 3 depicts one example of logical partition groups;
FIGs. 4a-4b depict one example of the logic associated with a partition joining a group;
FIG. 5 depicts one embodiment of the logic associated with removing a partition from a group;
FIG. 6 depicts one embodiment of the logic associ-

ated with determining if a partition's weight can be increased to help a receiver service class of the partition; and

FIG. 7 depicts one embodiment of the logic associated with dynamically adjusting the configuration of logical processors.

[0013] Workload management capabilities are provided that enable the dynamic adjustment of the allocation of resources of a computing environment to balance the workload of that environment. In one example, the computing environment includes a plurality of logical partitions and the workload is managed across two or more of the partitions.

[0014] One embodiment of a computing environment using workload management capabilities is described with reference to FIG. 1a. A computing environment 100 is based, for instance, on the Enterprise Systems Architecture (ESA)/390 offered by International Business Machines Corporation, Armonk, New York. ESA/390 is described in an IBM publication entitled "Enterprise Systems Architecture/390 Principles Of Operation," IBM Publication No. SA22-7201-04, June 1997. One example of a computing environment based on ESA/390 is the 9672 Parallel Enterprise Server offered by International Business Machines Corporation.

[0015] Computing environment 100 includes, for example, a central processor complex (CPC) 102 having one or more central processors 106 (e.g., CP1-CP4), one or more partitions 108 (e.g., logical partitions (LP1-LP4)), and at least one logical partition manager 110, each of which is described below.

[0016] Central processors 106 are physical processor resources that are allocated to the logical partitions. In particular, each logical partition 108 has one or more logical processors (not separately shown for clarity), each of which represents all or a share of a physical processor 106 allocated to the partition. The logical processors of a particular partition 108 may be either dedicated to the partition (so that the underlying processor resource 106 is reserved for that partition) or shared with another partition (so that the underlying processor resource is potentially available to another partition).

[0017] In the particular example shown, each of logical partitions LP1-LP4 functions as a separate system having a resident operating system 112 (which may differ for each logical partition) and one or more applications 114. In one embodiment, operating system 112 is the OS/390™ or MVS/ESA™ operating system offered by International Business Machines Corporation.

[0018] Additionally, each operating system (or a subset thereof) includes a workload manager 116 for managing the workload within a partition and among partitions. One example of a workload manager is WLM offered by International Business Machines Corporation. WLM is described in, for instance, U.S. 5,473,773, issued December 5, 1995; and U.S. 5,675,739, issued October 7, 1997.

[0019] Logical partitions 108 are managed by logical partition manager 110 implemented by microcode running on processors 106. Logical partitions 108 (LP1-LP4) and logical partition manager 110 each comprise one or more programs residing in respective portions of central storage associated with the central processors. One example of logical partition manager 110 is PR/SM.

[0020] In a further embodiment of a computing environment, two or more central processor complexes are coupled to one another to form a sysplex, as depicted in FIG. 1b. As one example, a central processor complex (CPC) 102 is coupled to one or more other CPCs 120 via, for instance, a coupling facility 122.

[0021] In the example shown, CPC 120 includes a plurality of logical partitions 124 (e.g., LP1-LP3), which are managed by a logical partition manager 126. One or more of the logical partitions includes an operating system, which may have a workload manager and one or more application programs (not shown in this example for clarity). Additionally, CPC 120 includes a plurality of central processors 128 (e.g., CP1- CP3), the resources of which are allocated among the plurality of logical partitions. In particular, the resources are allocated among one or more logical processors 130 of each partition. (In other embodiments, each CPC may have one or more logical partitions and one or more central processors.)

[0022] Coupling facility 122 (a.k.a., a structured external storage (SES) processor) contains storage accessible by the central processor complexes and performs operations requested by programs in the CPCs. The coupling facility is used for the sharing of state information used in making shared resource redistribution decisions. (In one embodiment, each central processor complex is coupled to a plurality of coupling facilities.) Aspects of the operation of a coupling facility are described in detail in such references as Elko et al., U.S. 5,317,739, issued May 31, 1994; U.S. 5,561,809, issued October 1, 1996; U.S. 5,706,432, issued January 6, 1998; and the patents and applications referred to therein.

[0023] In one embodiment, one or more of the central processors are coupled to at least one channel subsystem, which is used in communicating with I/O devices. For example, a central processor 200 (FIG. 2) is coupled to main storage 202 and at least one channel subsystem 204. Channel subsystem 204 is further coupled to one or more control units 206. The control units are then coupled to one or more I/O devices 208.

[0024] The channel subsystem directs the flow of information between the input/output devices and main storage. It relieves the central processing units of the task of communicating directly with the input/output devices and permits data processing to proceed concurrently with input/output processing. The channel subsystem uses one or more channel paths 214 as com-

munication links in managing the flow of information to or from input/output devices 208.

[0025] Each channel path 214 includes, for instance, a channel 210 of channel subsystem 204, a control unit 206 and a link 212 between the channel and control unit. In other embodiments, a channel path may have multiple channels, control units, and/or links. Further, in another example, it is also possible to have one or more dynamic switches as part of the channel path. A dynamic switch is coupled to a channel and a control unit and provides the capability of physically interconnecting any two links that are attached to the switch. Further details regarding channel subsystems are described in U.S. 5,526,484, issued on June 11, 1996.

[0026] In a preferred embodiment of the present invention, various physical resources are dynamically redistributed across the logical partitions of a computing environment under direction of one or more workload managers. This dynamic redistribution is transparent to the application subsystems. As examples, the physical resources to be redistributed include CPU resources, logical processor resources, I/O resources, coprocessors, channel resources, network adapters, and memory resources. As one example, a coprocessor is a microprocessor (other than a CPU) within a CPC that serves a particular function. Examples of coprocessors include, for instance, channel subsystems, network adapter cards and cryptographic coprocessors. The above physical resources are only offered as examples; other shareable resources may also be redistributed.

[0027] In order to facilitate the dynamic redistribution of resources, in one embodiment, logical partitions are grouped together in order to share the resources among the partitions of the group. Each group can vary in size from 1 partition to n partitions. (In one example, one or more of the groups include one or more partitions, but less than all of the partitions of the computing environment.) In particular, each group includes, for instance, one or more operating system images running in independent domains of a machine, which are managed by a common workload manager function to distribute workloads and resources. In one example, these domains are logical partitions running in logically partitioned mode and the operating systems are OS/390 running in the logical partitions. The logical partitions of a group may be a subset of the partitions of a system (e.g., a CPC) or a sysplex, an entire system or sysplex, or may be partitions of different sysplexes (on, for example, a single CPC) or systems.

[0028] One embodiment of two logical partition groups (or clusters) of a central processor complex is depicted in FIG. 3. As shown, there is a Logical Partition Group A 300 and a Logical Partition Group B 302, each of which includes one or more logical partitions. The grouping of logical partitions enables resource sharing among the partitions of a group through resource allocation (e.g., priority based resource allocation).

[0029] As examples, the resources to be shared include CPU resources, I/O resources, and memory, as well as co- processors or any other shareable resources the machine might provide. A particular logical partition group may or may not have access to all of the resources of a particular machine. In fact, multiple logical partition groups could be defined to operate concurrently on a single machine. In order to manage each logical partition group effectively, the resources that make up a particular logical partition group are effectively scoped to that group.

[0030] The scoping includes identifying which resources are allocatable to each group. In particular, the scope defines which resources are restricted to the group and can be managed for the group. The logical partitions that make up a logical partition group can be thought of as a container of the resources. These containers exist within the bounds of a total set of resources available to the logical partitions. In one example, this is the total set of resources available on a particular CPC.

[0031] The logical partitions that make up a particular logical partition group (e.g., Logical Partition Group A) are assigned a particular portion of the total shareable resource. For example, assume that the shareable resource is a CPU resource. With shared CPU resources, the logical partitions that are included in Logical Partition Group A are assigned a particular portion of the total central processor complex CPU resource. These resources are being shared by the logical partitions within a particular group, as well as, potentially, with logical partitions in other logical partition groups and logical partitions not included in any logical partition groups. Thus, a workload manager that is trying to make decisions about moving resources within a group (from, for instance, one partition in the logical partition group to another partition in the group) is to have an understanding of the resources that comprise the group, as well as an understanding of what the larger container (e.g., the CPC) contains. Measurement feedback (e.g., state information stored in the coupling facility) used to make decisions about managing workload resources should be sufficient to understand the customer defined containers as above.

[0032] Once this understanding is established, workload manager directed changes to the resource allocations in the logical partitions of a given group are typically done in such a way that keeps the container size (i.e., the resources allocated to the logical partition group) constant. For instance, assume that the resource to be managed is the CPU resource, and further assume that each logical partition is assigned a CPU processing weight that indicates priority. In order to manage the CPU relative weights, the sum of the relative weights for the logical partitions in a given group are to remain constant before and after the directed change, via, for instance, workload manager. This maintains the customer specified allocation of the resources to the groups and other logical partitions present on the machine.

[0033] Notwithstanding the above, in some cases it may be desirable and possible for the group of partitions to utilize resources greater than the defined container, when those resources are not being used by their designated owners. However, as soon as contention for the resources occurs, the resources are managed by the logical partition (LPAR) manager according to the defined container sizes (e.g., processing weights in this example). There may, however, be other cases when the group should not be allowed to expand beyond its container. This is also possible with scoping. Other resources may need to be fully scoped to a single group in order to get an accurate picture of usage of the resources. Limiting in this fashion prevents logical partitions outside of a given group from accessing that resource.

[0034] In addition to the above, consideration is also given to the effect of external changes on the availability of resources within a logical partition group. For example, a user may change the allocation of resources via some external means (not under workload manager direction). This might be done because of a change in actual workloads that are on a machine or a shift in business priorities between groups and/or other logical partitions. When these changes are made, these changes are to be understood by the workload manager and the effects of these changes are to be distributed rationally. Changes might occur when a logical partition is added to or removed from a group; when some other logical partition outside the group is added or removed; or simply, when a processing weight change is effected via external means. When these external changes are performed, the size of the container can change and workload manager is now the manager of that newly sized container.

[0035] When resources attributed to a particular logical partition of a group are changed externally, a redistribution of resources within a group may be needed. For instance, when a logical partition is removed from a group, the processing weight associated with that logical partition is removed from the group. If the current workload manager assigned weight for the logical partition is greater than the logical partition's weight that is being removed (i.e., the processing weight associated with the logical partition initially), the difference in weight is added to other logical partitions in the group. This is done, for instance, in proportion to the existing distribution of weights in the other logical partitions in the group. If the current workload manager assigned weight for the logical partition is less than the logical partition's initial weight, the difference in weight is subtracted from the other logical partitions in the group. Again, this is done in proportion to the other logical partition weight assignments, as one example.

[0036] As described above, a group is scoped in order to obtain a handle on the resources that are assigned to a group and the resources that are allowed to change, so that the workload manager can make proper decisions of what to do next. The scoping identifies the groups and provides information back to the program that the program can understand. When a group is modified, the resources are dynamically adjusted to satisfy the modification.

[0037] In one embodiment, there can be separate groups (clusters) for each resource. For example, Logical Partition Group A may be for CPU resources, while Logical Partition Group B may be for I/O resources. However, in other embodiments, it is also possible that one logical partition group is for a subset or all of the resources.

[0038] In order to establish LPAR group scope, in one example, the logical partitions identify themselves to one or more groups of partitions. One embodiment of the logic associated with joining a group is described with reference to FIGs. 4a-4b. For example, to join a logical partition group, the operating system (e.g., OS/390) running in a logical partition indicates to the LPAR manager which LPAR group the logical partition is to be a part thereof, STEP 400. As one example, an instruction is used to pass the LPAR group name to the LPAR manager. The operating system specifies a name for each type of resource that is to be managed within LPAR groups. Thus, if there are other resources, INQUIRY 402, then other names are specified. For example, a group name is given for CPU resources and another name is given for I/O resources. The same LPAR group name can be specified for each resource type, if desired.

[0039] This declaration by OS/390 either establishes a new LPAR group on the machine (if the logical partition is the first logical partition to use that name) or causes this logical partition to join an existing LPAR group of the same name for that resource type. For example, once the group name is specified, STEP 404 (FIG. 4b), a determination is made as to whether it is a new name, INQUIRY 406. If so, a new group is created, STEP 408.

[0040] Otherwise, an existing group is joined, STEP 410. Thereafter, the resources are scoped to the group, STEP 412.

[0041] In particular, the resources of the group type that are bound to the LPAR group are now made available for that logical partition to utilize, if and when WLM running in the LPAR group determines it should. The resources of a particular type for an LPAR group that need scoping include at least two varieties: additive resources and fixed resources.

[0042] Additive Resources: In some cases, joining an LPAR group inherently adds resources to the LPAR group that the logical partition just joined. An example of this is CPU processing weight, which is, for example, assigned by the customer to a logical partition at a hardware console. The current (in use) processing weight of the logical partition is initialized from this customer assigned weight, when the logical partition is activated. When the logical partition joins an LPAR group for CPU resources, the customer assigned processing weight for that logical partition becomes part of the total processing

weight available for use within the LPAR group, and thus, can be reassigned within the LPAR group by WLM. The logical partition that just joined the LPAR group now has the potential to use the larger set of LPAR group resources, to which a contribution was just made.

[0043] Fixed Resources: In some cases, a set of resources is predefined as belonging to a particular LPAR group. An example of this is managed (floating) channel paths. A managed channel path is a channel path whose resources can be reassigned to help achieve workload goals. The set of managed channel paths for use by a particular LPAR group is initially defined via an I/O configuration definition process that associates channel paths (CHPIDs) with an LPAR group. When a logical partition joins this LPAR group, it is now allowed access to this set of channel paths. The logical partition itself did not contribute anything to this resource pool. (This pool of resources can still be changed dynamically, but the point is that the resources do not come and go as logical partitions join and leave an LPAR group.)

[0044] LPAR scope can also be enforced differently for resources depending on the type of resource.

[0045] Additive Resources: The operating system in an LPAR group is to be able to query the complete set of resources of this type for the LPAR group. As an example, for CPU processing weights, this is accomplished via an instruction. The operating system learns the total set of this resource type within the LPAR group, the allocation of the resources to the logical partitions in the group, and the complete size of the resource pool available on the current machine. All these components are used to understand how much of the total physical resource is to be allocated to a logical partition. The operating system then updates the allocations to the logical partitions in the LPAR group to reassign resources within the group. The operating system is not allowed, in one example, to change the total amount of resource allocated to the LPAR group. The LPAR manager enforces this by making sure that all parts of the LPAR group are accounted for in an update and no logical partitions outside the LPAR group have their resources affected.

[0046] Fixed Resources: The operating system in an LPAR group queries the set of resources that is associated with its LPAR group for this type of resource. For example, with managed channel paths, a list of managed channel paths that are defined for a particular LPAR group can be retrieved from the LPAR manager via an instruction. The LPAR manager also screens these resources to make sure they are only utilized by the proper LPAR group. For managed channels, this means only allowing a managed channel path to be configured online to a logical partition that has declared an LPAR group name that matches that defined for the managed channel path.

[0047] When a logical partition that is part of an LPAR group is system reset, re-IPLed, or deactivated, any affiliation that logical partition had with one or more LPAR groups is removed. One embodiment of the logic associated with removing a logical partition from a group is described with reference to FIG. 5. As part of the reset, the logical partition manager removes a declared LPAR partition group name(s) from the logical partition, STEP 500. Then, one or more other actions are performed to complete the LPAR group resource deallocation for the logical partition, depending on the resource, INQUIRY 502.

[0048] If the resource is an additive resource, then the following occurs: resources such as this, that were added into an LPAR group when a logical partition joined the LPAR group, are removed from the LPAR group, STEP 504. This may involve an adjustment in the current allocation of this type of resource to the remaining members of the LPAR group. For instance, in the case of processing weights, the initial processing weight for the logical partition leaving the group is now removed from the scope of the LPAR group. If WLM had changed the current processing weight for the logical partition, adjustments need to be made. If the logical partition's current processing weight is greater than its initial processing weight, the difference between the two is redistributed to the remaining LPAR group members in proportion to their current processing weights. If the logical partition's current processing weight is less than its initial processing weight, the difference between the two is removed from the remaining LPAR group members in proportion to their current processing weights. The result of these adjustments re-establishes the processing weight container for the resulting LPAR group.

[0049] On the other hand, if the resource is a fixed resource, then the following occurs: resources such as these are simply removed from the configuration of the logical partition being reset, STEP 506. For instance, for managed channel paths, the channel paths are deconfigured from the logical partition being reset. This once again re-establishes that only members of the LPAR group have access to the LPAR group resource.

[0050] It should also be noted that some resources managed by WLM in an LPAR group environment may not have a need for group scoping. One example of such a resource is the number of logical central processors (CP) online for a logical partition. The effective behavior of a particular logical partition in an LPAR group can be significantly influenced by the number of logical CPs that are online to the logical partition. The number of logical CPs that a logical partition can have defined and/or online is a characteristic of a logical partition whether or not is it is in an LPAR group, so this resource does not really become part of a larger pool of resources. Its effect in an LPAR group though is that it can change what type of workload can effectively be run in one LPAR group member versus another.

[0051] In one example, a resource to be shared among a plurality of logical partitions is a CPU resource. The OS/390 workload manager redistributes CPU resources across logical partitions by dynamically adjust-

ing one or more relative processor weights associated with the logical partitions. WLM understands when an important workload is delayed because the weight of the partition it is running within is too low. WLM can help this workload by raising the weight of this partition and lowering the weight of another partition, thereby providing additional CPU capacity to the important workload. CPU resources dynamically move to the partitions where they are needed, as workload requirements change.

[0052] In one embodiment, the scope of WLM management of logical partition weights is a logical partition group. As one example, WLM adjusts logical partition weights, but maintains the sum of the weights of the partitions in the group constant. Maintaining the sum constant, keeps the overall CPU resource allocated to the group the same relative to other independent groups on the same physical computer. Therefore, when WLM raises the weight of one partition, it lowers the weight of another partition in the same group.

[0053] The management of logical partition weights is an enhancement to WLM's goal oriented resource allocation techniques, which are described in, for instance, U.S. 5,473,773, issued December 5, 1995; and U.S. 5,675,739, issued October 7, 1997.

[0054] As described in those patents, WLM controls the allocation of CPU resources within a logical partition by adjusting CPU dispatching priorities. CPU dispatching priorities are assigned to work at a service class level. However, there are various situations in which the adjustment of dispatching priorities does not help the service class. For example:

1) The service class is already alone at the highest CPU dispatching priority allowed to non-system work.

2) Changing CPU dispatching priorities to help the service class will have too large a negative impact on other service classes that are of equal or higher importance.

[0055] Thus, when WLM finds that a service class is missing its goals due to CPU delay, which cannot be helped by adjusting CPU priorities, WLM considers adjusting the weight of the partition associated with the failing service class.

[0056] The service class to which WLM is considering allocating additional resources is called the receiver service class. When WLM has found a receiver service class missing goals due to CPU delay on a given partition that cannot be helped for one of the reasons listed above, WLM considers raising that partition's weight. One embodiment of the logic followed by WLM to determine if a partition's weight can be increased to help the receiver service class is described as follows, with reference to FIG. 6:

1. Project the effect on the receiver class of increasing the weight of the partition, STEP 600. Increasing the partition's weight increases the CPU capacity of the partition. Since the CPU demand of the work in the receiver class is assumed to be constant, increasing the CPU capacity of the partition decreases the percentage of this capacity that the receiver service class demands. The projection of the benefit to the receiver service class is based on this decrease in the percentage of available CPU capacity both the receiver service class and the other work on the system demand.

2. Find another partition in the logical partition group to be a candidate to have its weight lowered, STEP 602. This partition is known as the candidate donor partition. The candidate donor partition is chosen by, for instance, looking for the partition where the least important work is likely to be impacted by lowering the partition's weight.

3. Project the effect on all service classes with work running on the candidate donor partition of lowering its weight, STEP 604. Decreasing the candidate donor partition's weight decreases the CPU capacity of the candidate donor partition. This decrease in CPU capacity means that the CPU demand of the service classes with work running on the candidate donor as a percentage of the capacity of the candidate donor will increase. The projection of the negative effect of reducing the candidate donor's weight is based on this increase in the percentage of available CPU capacity that these service classes demand.

4. Determine if this change in weight has net value, INQUIRY 606. That is, the benefit to the receiver service class overrides the negative impact to work on the candidate donor partition based on the goals and importance of the service classes involved.

5. If adjusting the weights does have net value, implement the proposed change to the partition's weights, STEP 608. If there is no net value, then a determination is made as to whether there are more candidate donor partitions, INQUIRY 610. If so, another candidate donor partition is chosen, STEP 612, and processing continues at step 3, STEP 604. If there are no more candidate donor partitions, then processing ends, STEP 614.

[0057] To enable WLM running on one partition to make a projection on the effect of changing partition weights on work running on another partition, each partition has access to a shared data structure containing performance data about each logical partition in the group. This partition level performance data includes, for instance:

- CPU requirements of work running on the partition by service class;
- How well each service class is doing towards its goal on the partition;
- CPU usage by CPU dispatching priority for the partition.

[0058] In a preferred embodiment of the present invention implented in an OS/390 system, this shared data structure is built and maintained in a coupling facility. However, other data sharing approaches could be used to implement this data structure, such as messaging or shared disk.

[0059] Described above is a capability for dynamically redistributing CPU resources of a computing environment. The resources are redistributed across logical partitions, as one example, by dynamically adjusting logical partition weights.

[0060] In addition to dynamically adjusting CPU resources of a computing environment, logical processor resources may also be dynamically adjusted.

[0061] A logical partition is configured with one or more logical processors, which are dispatched on the central processor complexes' physical central processing units to execute work. To allow a partition to consume its assigned CPU capacity, sufficient logical processors are to be configured to the logical partition. For example, consider the case of Logical Partition A running on a CPC with ten CPUs. If a workload manager assigns Logical Partition A 50% of the CPC's capacity, Logical Partition A needs at least five logical processors to be configured to it. (Five logical processors could run on five of the CPUs or 50% of the CPC's capacity.) Should Logical Partition A later be assigned 95% of the CPC's capacity, Logical Partition A would then be configured with ten logical processors. Since WLM can dynamically adjust the capacity assigned to Logical Partition A with a statically defined logical processor configuration, ten logical processors are configured to Logical Partition A in order to accommodate all possible capacity assignments. However, should Logical Partition A be assigned, for example, only 20% of the CPC's capacity, two problems arise from the statically defined logical processors: 1) Each of the ten logical processors will, on average, be allowed to consume physical CPU resources at the rate of only 0.2 of a physical CPU's capacity (20% of ten CPUs divided by ten logical processors equals 0.2 CPUs per logical processor). This can severely restrict workloads whose throughput is gated by a single task, since that single task will only be able to execute at 0.2 the capacity of the physical CPU - this is often referred to as the short engine effect; 2) Software and hardware efficiency is significantly reduced when having to manage ten logical processors, when only two logical processors are required.

[0062] In order to address the above deficiencies, the configuration of a logical partition is not statically defined, but instead is dynamically adjusted. In one example, it is WLM that manages the partition and makes the dynamic adjustment. WLM can do this for each logical partition of a computing environment (or within an LPAR group). One embodiment of the logic associated with dynamic adjustment of the configuration of logical processors is described with reference to FIG. 7.

[0063] Initially, a logical partition is configured with the minimum number of logical processors required to allow it to consume the capacity assigned to the logical partition by workload manager (or the capacity actually being used, if larger), STEP 700. As the logical partition's capacity assignment (or capacity use) changes, INQUIRY 702, an evaluation is made to determine whether the number of logical processors configured to the logical partition should be altered, STEP 704. In one example, the number of logical processors configured to a logical partition remains close to the number of physical CPUs necessary to provide the CPC capacity assigned to (or used by) a logical partition. Thus, each logical processor executes at close to the capacity of a physical CPU and the number of logical processors to manage are minimized.

[0064] In order to make the evaluation of whether to change the logical configuration, the following equation is used in one example: L=floor[max(W,U)xP+1.5] subject to a maximum of L=P, where L=number of logical processors configured to a logical partition; W=percentage of CPC capacity assigned to the logical partition; U=percentage of CPC capacity currently being used by the logical partition; and P=number of physical CPUs on a CPC, STEP 705.

[0065] L is evaluated by workload manager based on the then current values of P,W and U at, for instance, regular and frequent intervals (e.g., every 10 seconds). Thresholds are used to determine if the actual value of L (L-act) for the logical partition should be raised or lowered. If the newly calculated value of L (L-calc)is higher than the current value of L-act, INQUIRY 706, then L-act is raised to L-calc, STEP 708. Otherwise, if L-calc is a value of two or more lower than L-act, INQUIRY 710, then L-act is set to L-calc minus one, STEP 712. If L-calc is equal to L-act or only a value of one below L-act, no change in the value of L-act for the logical partition is made, STEP 714. Through the use of these thresholds, unnecessary changes of L-act due to small workload fluctuations are avoided, while still being responsive to quickly increasing capacity demands of workloads.

[0066] For further illustration, consider the following example: Assume P=10, W=U=24%. A static configuration of logical processors would require L(static)=10 to handle the case should W grow to greater than 90%. However, in a preferred embodiment of the present invention: L(Dynamic)=floor[max(.24,.24)x10+1.5]=3. Thus, in this example, L(static) would constrain a single task to execute at 0.24 of a physical CPU, while L(Dy-

namic) allows a single task to execute at 0.80 of a physical CPU, thereby providing a 233% increase in throughput for workloads gated by single task performance. Additionally, the software and hardware efficiency is significantly improved since, in this example, only three logical processors are managed rather than the ten logical processors needed for L(static).

[0067] Described above are various mechanisms for managing resources of a computing environment. Physical shareable resources are managed across logical partitions of a computing environment. The logical partitions are grouped to enable resource sharing through, for instance, priority based resource allocations. This resource sharing includes, for example, dynamic management of CPU resources across LPARs; dynamic CHPID management across LPARs; I/O priority queueing in the channel subsystem; and dynamic management of memory across LPARs.

[0068] In the embodiments described above, various computing environments and systems are described. It will be appreciated that these are only examples and are not intended to limit the scope of the present invention. Likewise, the flow diagrams depicted herein are just exemplary. There may be many variations to these diagrams or the steps (or operations). For instance, the steps may where appropriate be performed in a differing order, or steps may be added, deleted or modified.

Claims

1. A method of managing logical processors of a computing environment including one or more logical partitions, said method comprising the steps of:

   configuring a logical partition of said computing environment with one or more logical processors; and
   dynamically adjusting the configuration.

2. The method of claim 1, wherein said dynamically adjusting is in response to the workload of said logical partition.

3. The method of claim 1 or 2, wherein said dynamically adjusting comprises the step of increasing the number of logical processors allocated to said logical partition.

4. The method of any preceding claim, further comprising the step of making a determination that said configuration is to be adjusted at a plurality of time intervals.

5. The method of claim 4, wherein said step of making a determination comprises the step of using a predefined equation in making the determination, the result of said predefined equation being compared with one or more thresholds to determine whether the adjustment is to be made.

6. The method of claim 5, wherein said predefined equation comprises:

   $$L=\mathrm{floor}[\max(W,U)*P+1.5],$$

   wherein: L=number of logical processors configured to said logical partition; W=percentage of central processor capacity assigned to said logical partition; U=percentage of central processor capacity currently being utilized by said logical partition; and P=number of physical processors that can be allocated on the central processor associated with said logical partition.

7. The method of claim 6, wherein said equation is subject to a maximum of L=P.

8. A computer program for implementing the method of any preceding claim.

9. A system for managing logical processors in a computing environment including one or more logical partitions, said system comprising:

   means for configuring a logical partition of said computing environment with one or more logical processors; and
   means for dynamically adjusting the configuration.

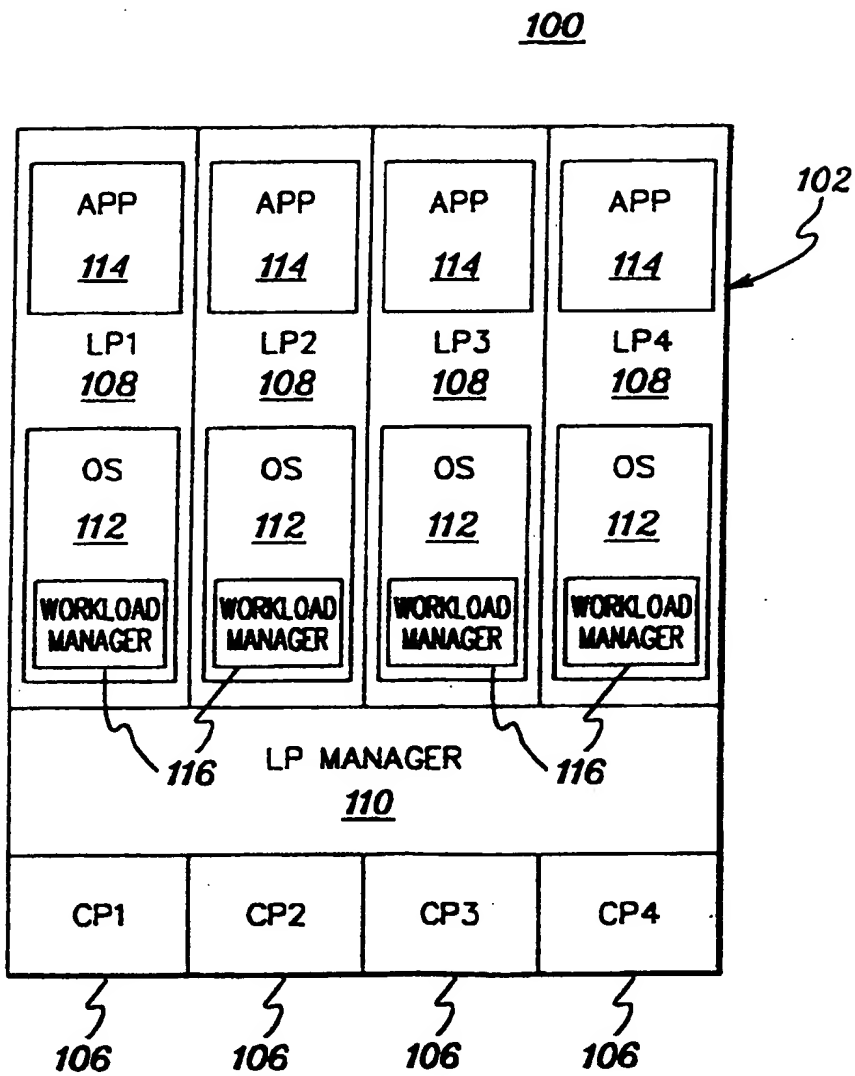
100
102
APP
114
APP
114
APP
114
APP
114
LP1
108
LP2
108
LP3
108
LP4
108
OS
112
OS
112
OS
112
OS
112
WORKLOAD MANAGER
WORKLOAD MANAGER
WORKLOAD MANAGER
WORKLOAD MANAGER
116
LP MANAGER
110
116
CP1
CP2
CP3
CP4
106
106
106
106

fig. 1A

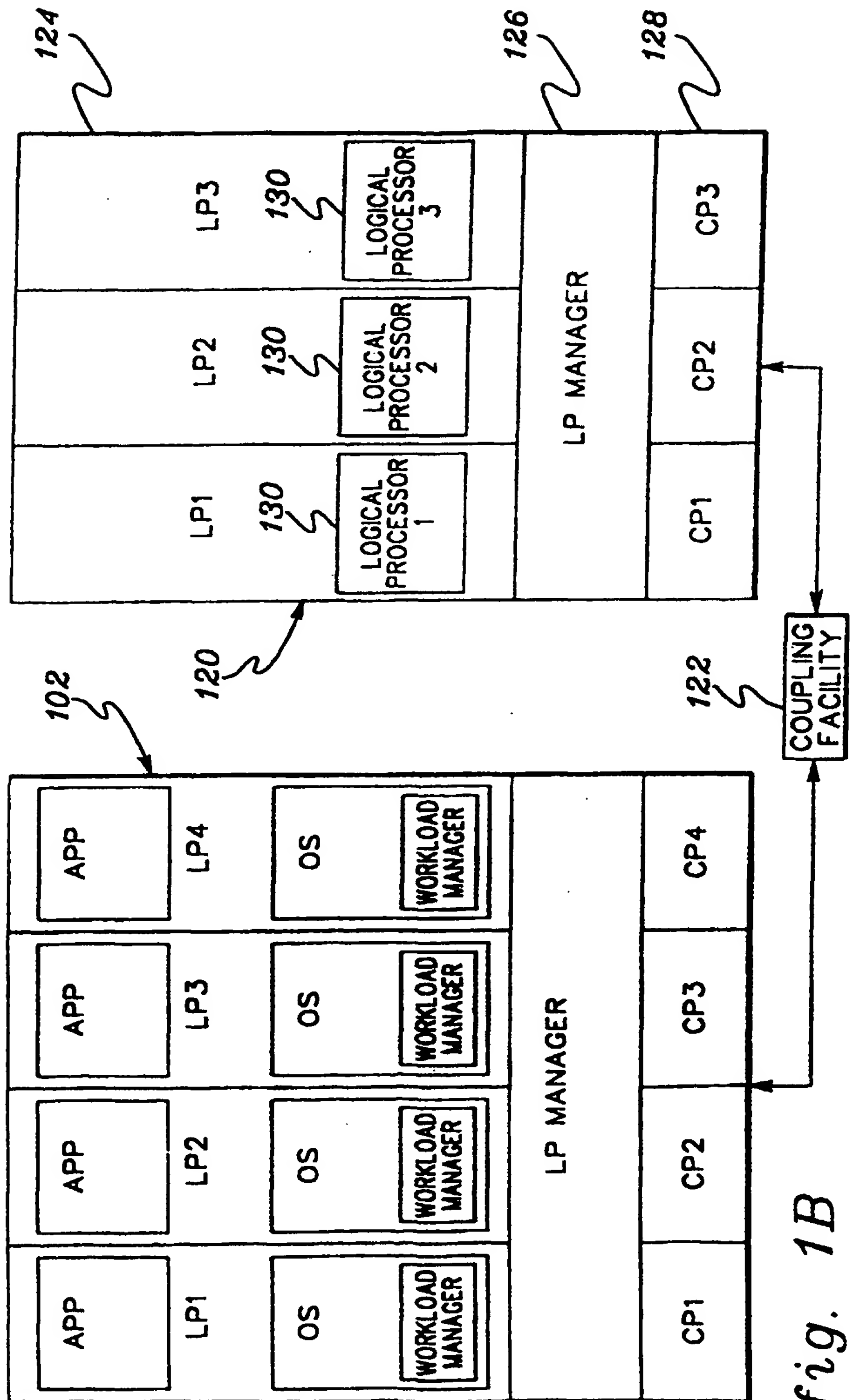
124
126
128
LP3
130
LOGICAL PROCESSOR 3
LP2
LOGICAL PROCESSOR 2
LP1
LOGICAL PROCESSOR 1
LP MANAGER
CP3
CP2
CP1
120
102
122
COUPLING FACILITY
APP
LP4
OS
WORKLOAD MANAGER
LP3
LP2
LP1
CP4

fig. 1B

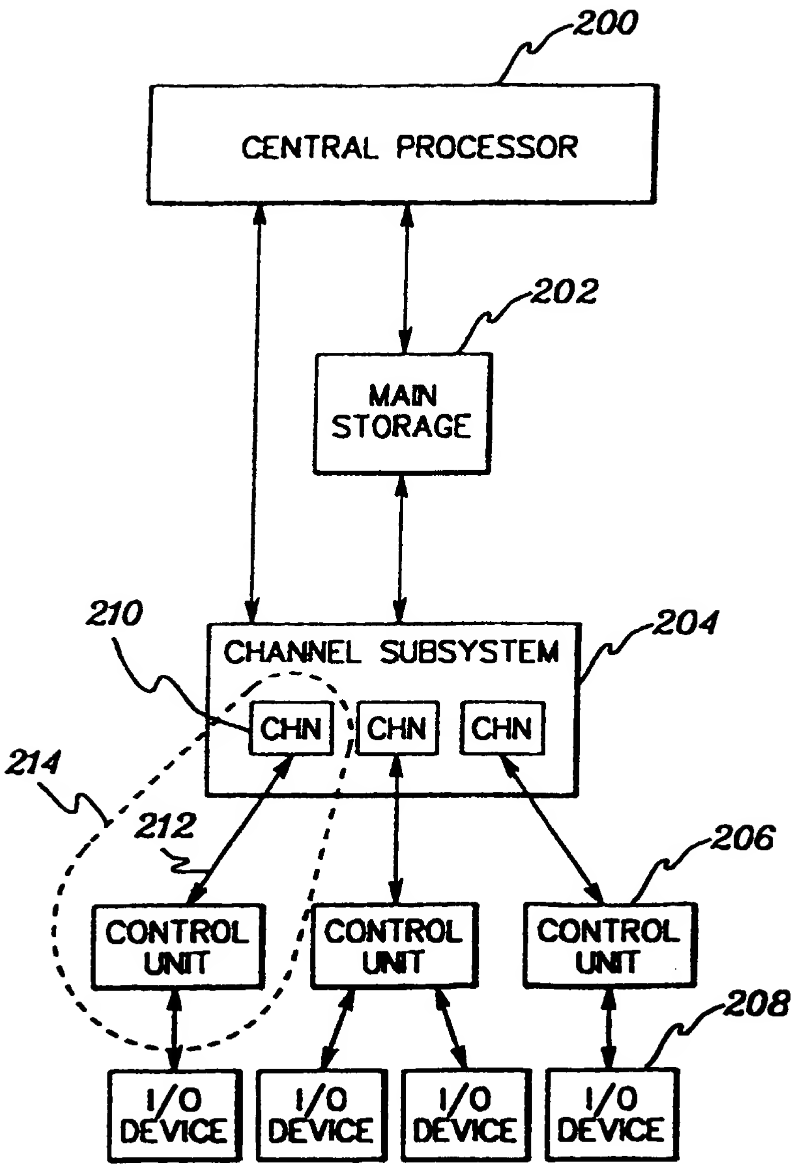
200
CENTRAL PROCESSOR
202
MAIN STORAGE
210
CHANNEL SUBSYSTEM
204
CHN
CHN
CHN
214
212
206
CONTROL UNIT
CONTROL UNIT
CONTROL UNIT
208
I/O DEVICE
I/O DEVICE
I/O DEVICE
I/O DEVICE

fig. 2

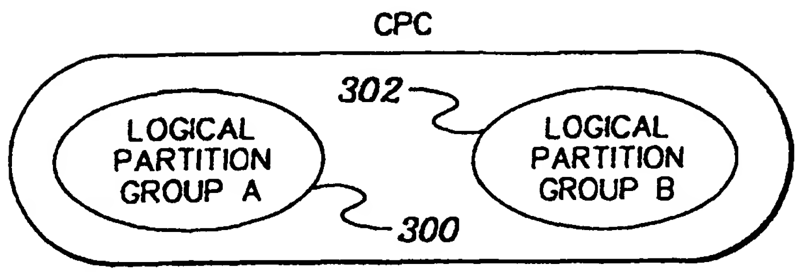
CPC
302
LOGICAL PARTITION GROUP A
LOGICAL PARTITION GROUP B
300

fig. 3

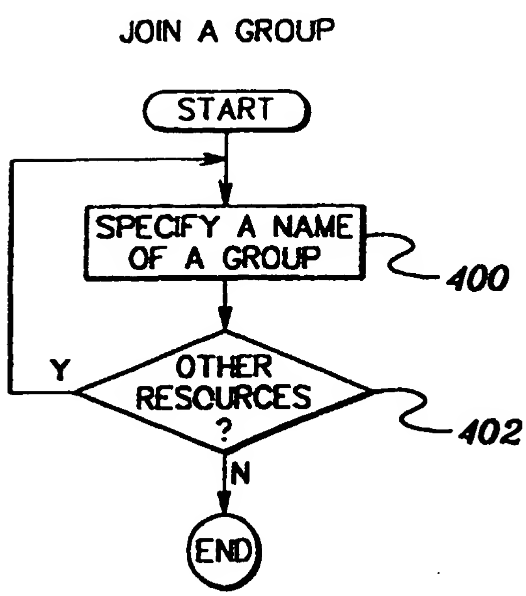
JOIN A GROUP
START
SPECIFY A NAME OF A GROUP
400
OTHER RESOURCES ?
Y
N
402
END

fig. 4A

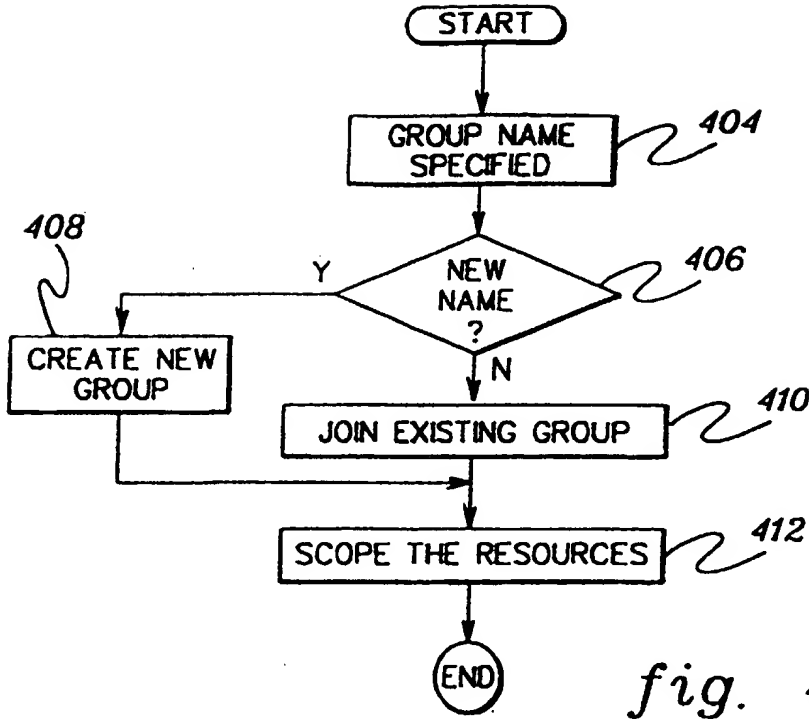

fig. 4B

REMOVE FROM A GROUP

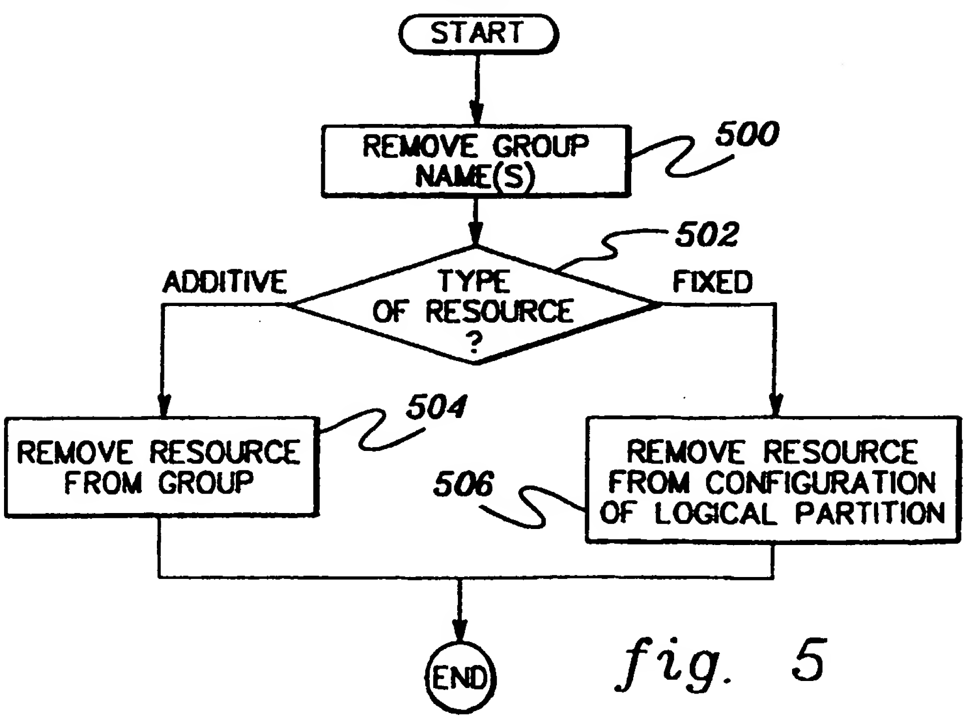

fig. 5

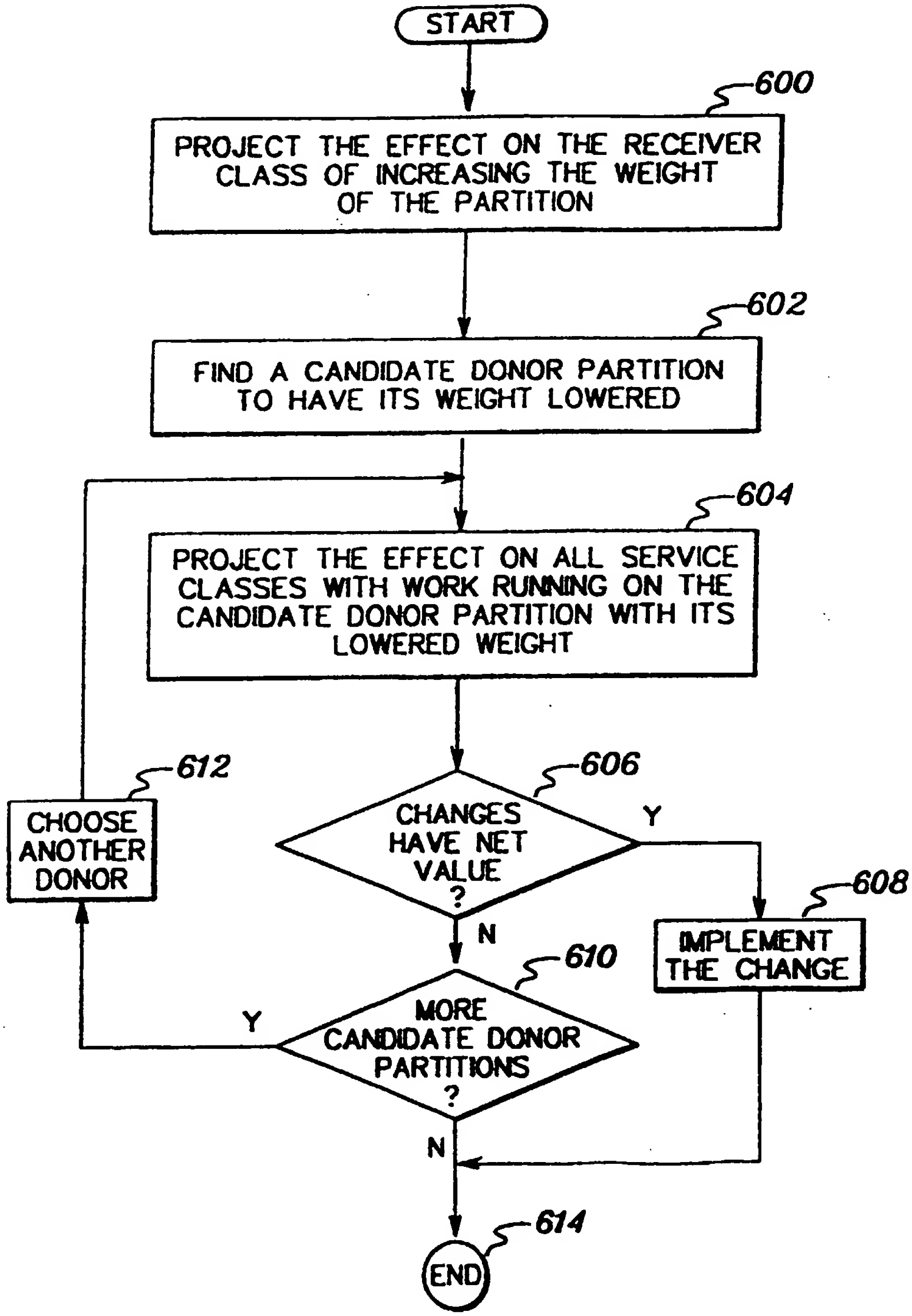
START
600
PROJECT THE EFFECT ON THE RECEIVER CLASS OF INCREASING THE WEIGHT OF THE PARTITION
602
FIND A CANDIDATE DONOR PARTITION TO HAVE ITS WEIGHT LOWERED
604
PROJECT THE EFFECT ON ALL SERVICE CLASSES WITH WORK RUNNING ON THE CANDIDATE DONOR PARTITION WITH ITS LOWERED WEIGHT
606
CHANGES HAVE NET VALUE ?
Y
608
IMPLEMENT THE CHANGE
N
610
MORE CANDIDATE DONOR PARTITIONS ?
Y
612
CHOOSE ANOTHER DONOR
N
614
END

fig. 6

DYNAMIC ADJUSTMENT OF CONFIGURATION OF LOGICAL PROCESSORS

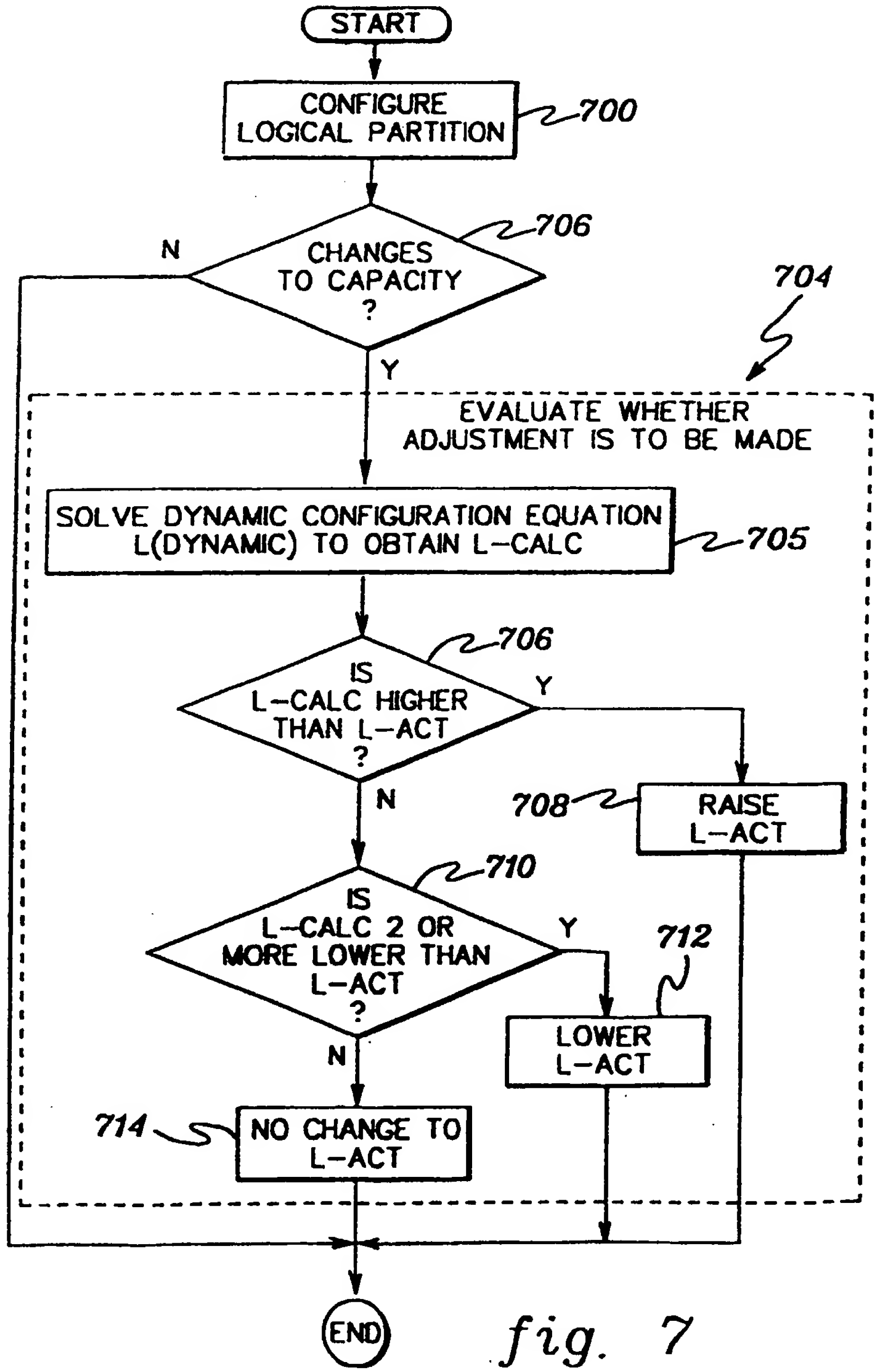

fig. 7